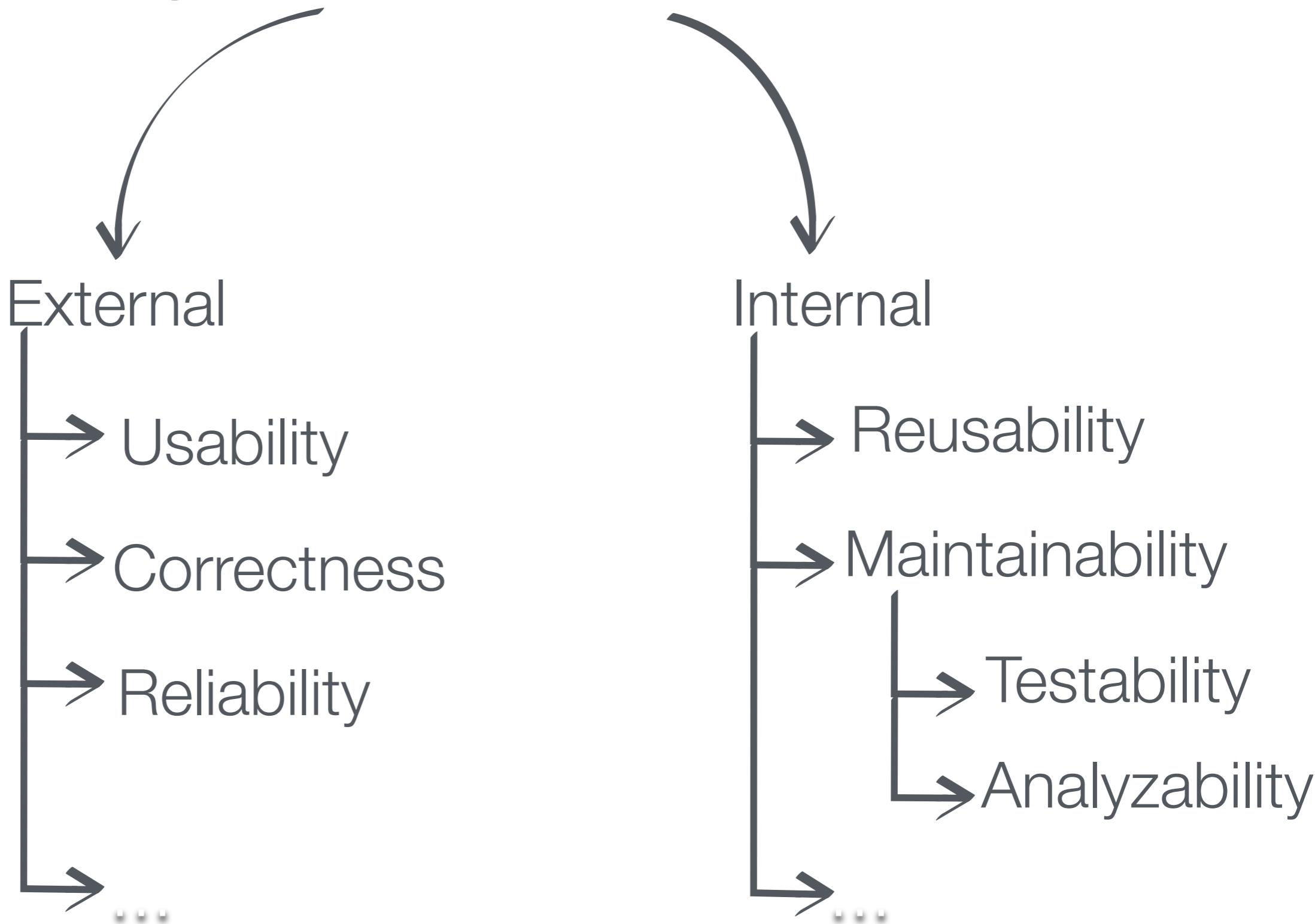


Tools for Improving Software Quality

WS 16-17



Assessing Software Quality Properties



Fostering Software Quality by Means of...

Constructive SQA | Analytical SQA

Programming Languages

Software Architecture

Scalable Static Analyses

Domain Specific Languages

Lightweight Formal Methods

Metrics

Software Development Processes

Machine Learning

Type Systems

(Language Based Security)

Integrated Development Environments

Continuous Delivery

- Unit/Acceptance-tests
- Code coverage and static analysis
- Deployment to integration environment
- Integration tests
- Deployments to Performance test environment
- Performance tests
- Alerts, reports and Release Notes sent out
- Deployment to release repository

Writing Tests / Assessing Tests

Writing Tests

- JUnit
- TestNG
- Hamcrest
(write tests by “matching objects”)
- ScalaTest
- ...

Code Coverage

- ECLEmma
- Cobertura
- scovrage
- ...

TestNG

```
// This method will provide data to any test method
// that declares that its Data Provider is named "provider1".
@DataProvider(name = "provider1")
public Object[][] createData1() {
    return new Object[][] {
        { "Cedric", new Integer(36) },
        { "Anne", new Integer(37) }
    };
}

// This test method declares that its data should be
// supplied by the Data Provider named "provider1".
@Test(dataProvider = "provider1")
public void verifyData1(String n1, Integer n2) {
    System.out.println(n1 + " " + n2);
}
```

Hamcrest

```
import static org.hamcrest.MatcherAssert.assertThat;
import static org.hamcrest.Matchers.*;

import junit.framework.TestCase;

public class BiscuitTest extends TestCase {
    public void testEquals() {
        Biscuit theBiscuit = new Biscuit("Ginger");
        Biscuit myBiscuit = new Biscuit("Ginger");
        assertThat(theBiscuit, equalTo(myBiscuit));
    }
}
```

ScalaTest

(also for testing Java!)

```
class DefaultIntegerRangesTest extends FunSpec with Matchers {  
    describe("IntegerRange values") {  
        describe("the behavior of the join operation") {  
            it("[0,0] join [1,Int.MaxValue] should result in [0,Int.MaxValue]") {  
                val v1 = IntegerRange(lb = 0, ub = 0)  
                val v2 = IntegerRange(lb = 1, ub = 2147483647)  
                v1.join(-1, v2) should be(StructuralUpdate(IntegerRange(0, 2147483647)))  
                v2.join(-1, v1) should be(StructuralUpdate(IntegerRange(0, 2147483647)))  
            }  
        }  
    }  
}
```

small concise tests

very good support for Pattern Matching

JaCoCo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
org.jacoco.examples		57%		69%	26	55	102	192	22	41	6	12
org.jacoco.agent.rt		80%		86%	32	109	59	275	24	72	7	20
jacoco-maven-plugin		87%		78%	30	142	43	347	5	86	0	17
org.jacoco.core		98%		99%	25	854	34	1,940	19	533	0	85
org.jacoco.report		99%		99%	6	542	5	1,286	2	362	0	65
org.jacoco.ant		98%		99%	5	157	10	419	4	109	0	19
org.jacoco.agent		85%		75%	3	11	5	30	1	7	0	1
Total	1,051 of 19,352	95%	56 of 1,259	96%	127	1,870	258	4,489	77	1,210	13	219

Code Coverage Report for JaCoCo 0.7.3-SNAPSHOT

Created with JaCoCo 0.7.3.201410310304

Measuring Code Coverage

JaCoCo (ECL Emma Team)

Build Tools

- (c)make
- ant
- maven
- sbt
- gradle
- ...



The interactive build tool
for Java Platform based projects

```
import AssemblyKeys._

name := "BugPicker"

version := "1.1.0"                                Version Information

scalaVersion := "2.11.4"

scalacOptions in (Compile, doc) := Seq("-deprecation", "-feature", "-unchecked", "-language:postfixOps")
scalacOptions in (Compile, doc) += Opts.doc.title("OPAL - BugPicker")                         Compiler Settings

libraryDependencies += "org.scalafx" %% "scalafx" % "1.0.0-R8"                                Project Dependencies

jfxSettings

JFX.addJfxrtToClasspath := true                     Project Settings

JFX.mainClass := Some("org.opalj.bugpicker.BugPicker")

assemblySettings

jarName in assembly := "bugpicker-" + version.value + ".jar"                               Deployment information

test in assembly := {}

mainClass in assembly := Some("org.opalj.bugpicker.BugPicker")

resourceGenerators in Compile <+= Def.task {
    val versionFile = (baseDirectory in Compile).value / "opalj" / "bugpicker" / "version.txt"
    versionFile.getParentFile.mkdirs()
    IO.write(versionFile, (version in Compile).value)
}                                                 Generation of other Artifacts
```

/ "classes" / "org" /

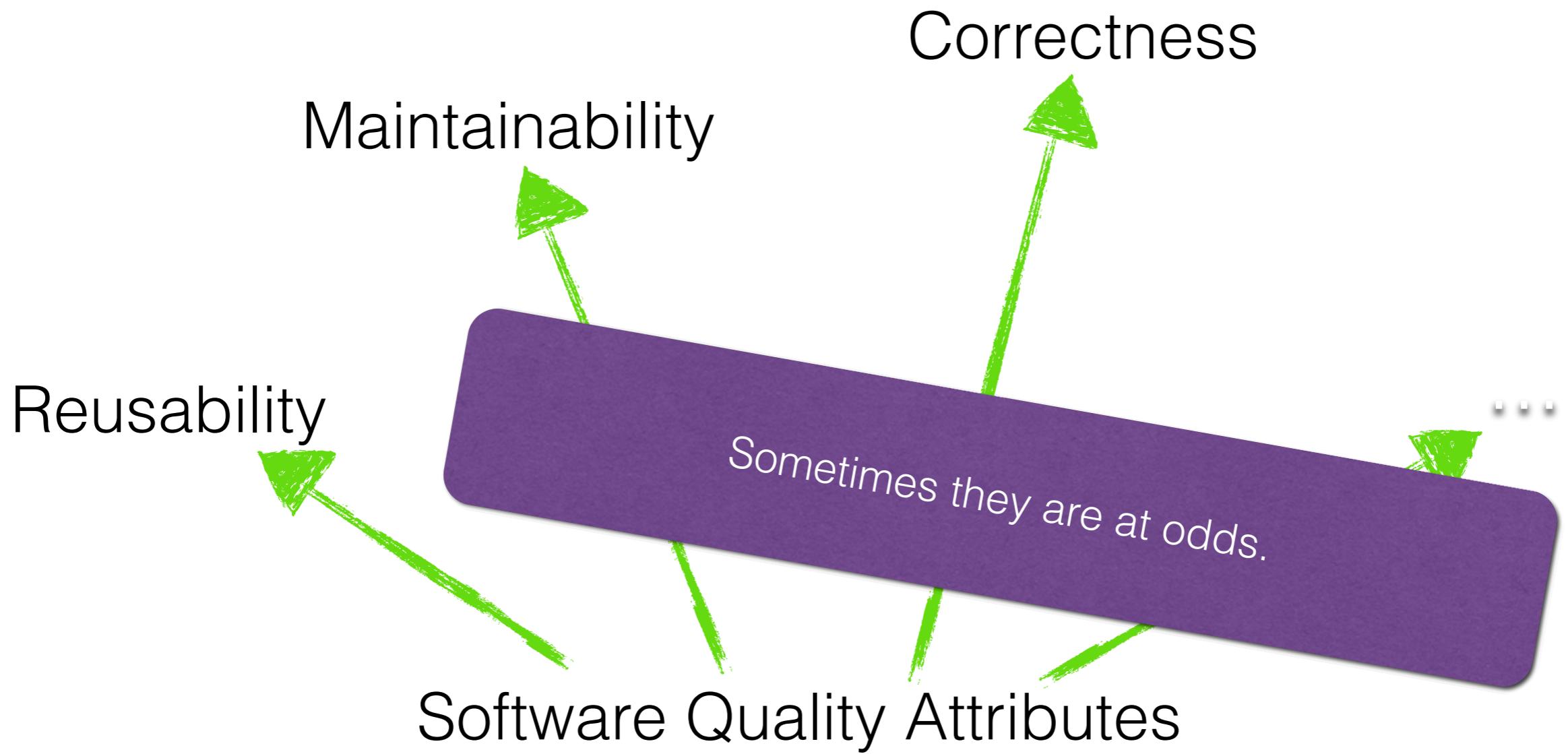
Easily hundreds of lines for larger projects.

Example Interaction

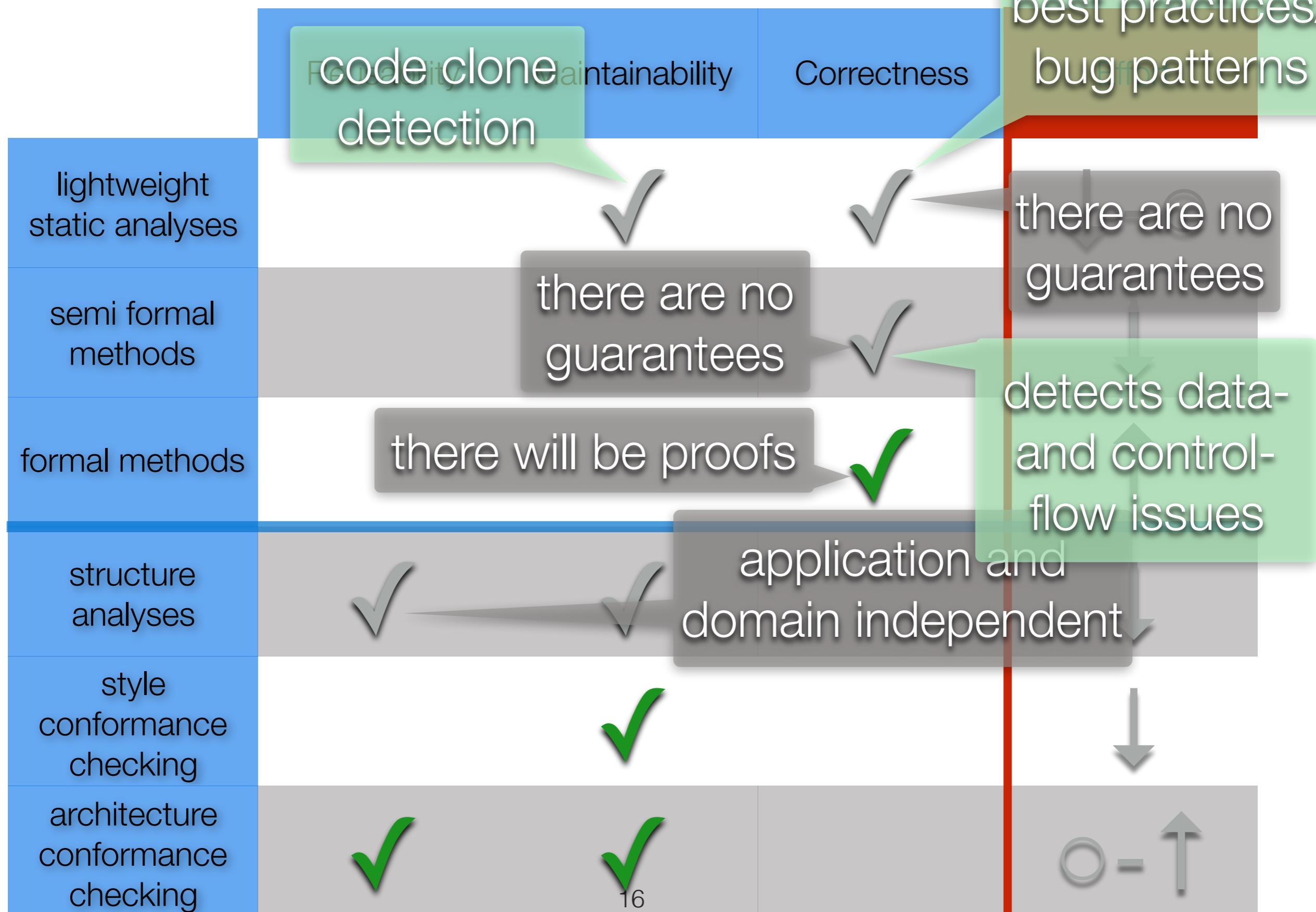
```
pc-eichberg-adapter:OPAL Michael$ sbt
[...]
[info] Set current project to OPAL Library (in build file:/Users/Michael/Code/OPAL/)
> project Common
[info] Set current project to Common (in build file:/Users/Michael/Code/OPAL/)
> clean
[...]
[success] Total time: 0 s, completed 20.11.2014 16:23:18
> test
[info] Compiling 3 Scala sources to /Users/Michael/Code/OPAL/OPAL/common/target/scala-2.11/test-
classes...
[info] LocalsTest:
[info] a Locals data structure
[info] - should be empty if it has size 0
[...]
[info] - the number of leaf nodes should be close to 1/4 of the number of entries
[info]   + for storing 10000 values 3008 nodes are required
[info] All tests passed.
[...]
[success] Total time: 7 s, completed 20.11.2014 16:23:15
> it:test
[...]
[success] Total time: 0 s, completed 20.11.2014 16:23:18
> doc
[...]
[success] Total time: 0 s, completed 20.11.2014 16:23:18
> publishLocal
[...]
[info] published ivy to /Users/Michael/.ivy2/local/de.opal-project/common_2.11/0.8.0-SNAPSHOT/ivys/
ivy.xml
[success] Total time: 0 s, completed 20.11.2014 16:23:43
>
```

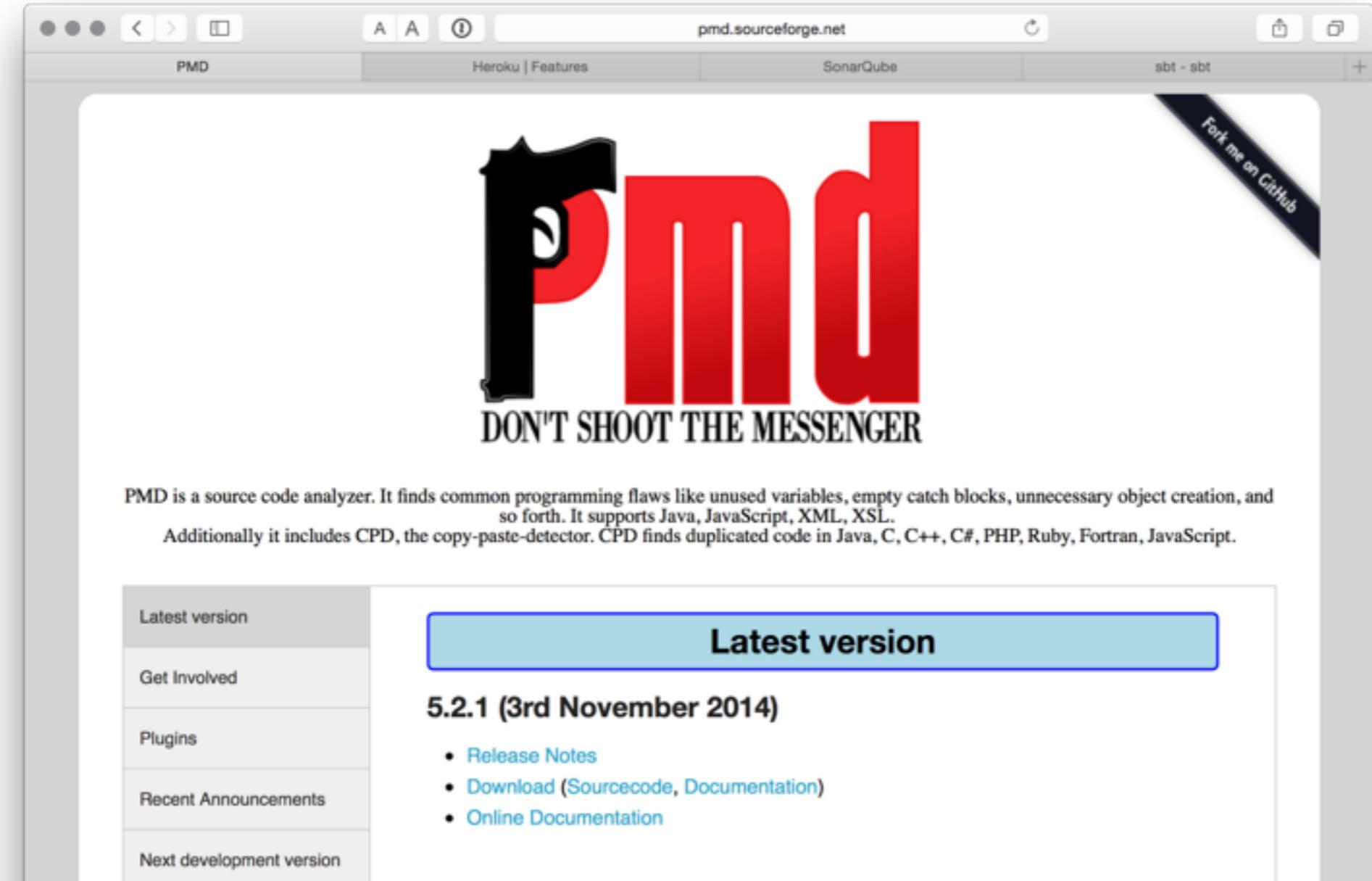
Static Analysis

Static Analyses are done to Improve Software Quality!



Static Analyses - Classified





PMD

A “simple” Java source code analyzer.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the file structure of the project, including various Java files like MONITOREXTRReader.java, NewArrayReader.java, etc., and several XML-related classes.
- Code Editor:** Displays the code for `ExecutionGraphBuilder.java`. The code includes methods for getting instruction memory layouts, referencing instructions, and target instructions, along with a constructor that builds the execution graph.
- Violations Outline:** A table showing violations:

Pri	Line	created	Rule
165		Thu Nov 20 16:52:09...	LawOfDemeter
280		Thu Nov 20 16:52:09...	LawOfDemeter
148		Thu Nov 20 16:52:09...	LawOfDemeter
151		Thu Nov 20 16:52:09...	LongVariable
275		Thu Nov 20 16:52:09...	ShortVariable
206		Thu Nov 20 16:52:09...	LawOfDemeter
57		Thu Nov 20 16:52:09...	CommentRequired
241		Thu Nov 20 16:52:09...	MethodArgumentCouldBeFinal
59		Thu Nov 20 16:52:09...	LooseCoupling
263		Thu Nov 20 16:52:09...	LawOfDemeter
60		Thu Nov 20 16:52:09...	LooseCoupling
68		Thu Nov 20 16:52:09...	BeanMembersShouldSerialize
265		Thu Nov 20 16:52:09...	PositionLiteralsFirstInComma
59		Thu Nov 20 16:52:09...	BeanMembersShouldSerialize
279		Thu Nov 20 16:52:09...	LongVariable
367		Thu Nov 20 16:52:09...	AvoidPrintStackTrace
110		Thu Nov 20 16:52:09...	NullAssignment
900		Thu Nov 20 16:52:09...	IssueIdentifier
- Violations Overview:** A table showing the distribution of violations across the project:

Element	# Violations	# Violations/KLOC	# Violations/Methc	Project
de.tud.bat.io.xml.reader.executiongraph	380	964.5	9.27	BAT2XML
ExecutionGraphBuilder.java	191	955.0	17.36	BAT2XML
LooseCoupling	7	35.0	0.64	BAT2XML
MethodArgumentCouldBeFinal	6	30.0	0.55	BAT2XML
AvoidInstantiatingObjectsInLoops	3	15.0	0.27	BAT2XML
CommentRequired	10	50.0	0.91	BAT2XML
DefaultPackage	2	10.0	0.18	BAT2XML
NullAssignment	2	10.0	0.18	BAT2XML
AvoidDuplicateLiterals	1	5.0	0.09	BAT2XML
UselessParentheses	2	10.0	0.18	BAT2XML
BeanMembersShouldSerialize	10	50.0	0.91	BAT2XML
ImmutableField	7	35.0	0.64	BAT2XML
EmptyIfStmt	4	20.0	0.36	BAT2XML

PMD

Defines more than ~300 rules.

Checkstyle 6.1



Last Published: 2014-11-13 | Version: 6.1

About

[Checkstyle](#)
[Release Notes](#)

Documentation

- ▼ Configuration
 - [Property Types](#)
- ▼ Running
 - [Ant Task](#)
 - [Command Line](#)
 - [Available Checks](#)
- ▼ Standard Checks
 - [Annotations](#)
 - [Block Checks](#)
 - [Class Design](#)
 - [Coding](#)
 - [Duplicate Code](#)
 - [Headers](#)
 - [Imports](#)
 - [Javadoc Comments](#)
 - [Metrics](#)
 - [Miscellaneous](#)
 - [Modifiers](#)
 - [Naming Conventions](#)
 - [Regexp](#)
 - [Size Violations](#)
 - [Whitespace](#)

Overview

Checkstyle is a development tool to help programmers write Java code that adheres to a coding standard. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard.

Checkstyle is highly configurable and can be made to support almost any coding standard. An example configuration files are supplied supporting the [Sun Code Conventions](#), [Google Java Style](#).

A good example of a report that can be produced using Checkstyle and [Maven](#) can be seen [here](#).

Important Development Changes

As of September 2013, the Checkstyle project is using GitHub for hosting the

Checkstyle

Automatically checking Code Conventions

The following document contains the results of Checkstyle  [XML](#)

Summary

Files	Info	Warnings	Errors
29173	0	0	546

Rules

Rule	Violations	Severity
SuppressionFilter <ul style="list-style-type: none">file: "suppressions.xml"	0	Error
SimpleJavadoc	546	Error

Details

Openjdk/Corba/Make/Tools/Src/Build/Tools/Stripproperties/StripPropertiesCorba.java

Severity	Message	Line
Info	1000000000000000 Mixed HTML class and C comments that class too mixed for	251

Checkstyle

Example Output

The screenshot shows a web browser window with the URL findbugs.sourceforge.net in the address bar. The page content is as follows:

FindBugs™ - Find Bugs in Java Programs

This is the web page for FindBugs, a program which uses static analysis to look for bugs in Java code. It is free software, distributed under the terms of the [Lesser GNU Public License](#). The name FindBugs™ and the [FindBugs logo](#) are trademarked by [The University of Maryland](#). FindBugs has been downloaded more than a million times.

The current version of FindBugs is 3.0.0.

FindBugs requires JRE (or JDK) 1.7.0 or later to run. However, it can analyze programs compiled for any version of Java, from 1.0 to 1.8.

The current version of FindBugs is 3.0.0, released on 20:43:52 CDT, 06 July, 2014. [We are very interested in getting feedback on how to improve FindBugs](#). File bug reports on [our sourceforge bug tracker](#).

[Changes](#) | [Talks](#) | [Papers](#) | [Sponsors](#) | [Support](#)

FindBugs 3.0.0 Release

- FindBugs supports Java 8 now (both as runtime and target platform).
- FindBugs requires minimum Java 7 as runtime environment!
- FindBugs uses ASM 5 now which means that some 3rd party detectors based on FindBugs 2.x/ASM 3 has to be upgraded. See details in [ASM documentation](#).
- New Bug patterns: [NP OPTIONAL RETURN NULL](#), [IIO INEFFICIENT INDEX OF](#), [IIO_INEFFICIENT_LAST_INDEX_OF_CNT_ROUGH_CONSTANT_VALUE](#)
- New "Source" filter which can be used to filter out classes generated from other languages:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Docs and Info

- [FindBugs 2.0](#)
- [Demo and data](#)
- [Users and supporters](#)
- [FindBugs blog](#)
- [Fact sheet](#)
- [Manual](#)
- [Manual\(ja/日本語\)](#)
- [FAQ](#)
- [Bug descriptions](#)
- [Bug descriptions\(ja/日本語\)](#)
- [Bug descriptions\(fr\)](#)
- [Mailing lists](#)
- [Documents and Publications](#)
- [Links](#)

Downloads

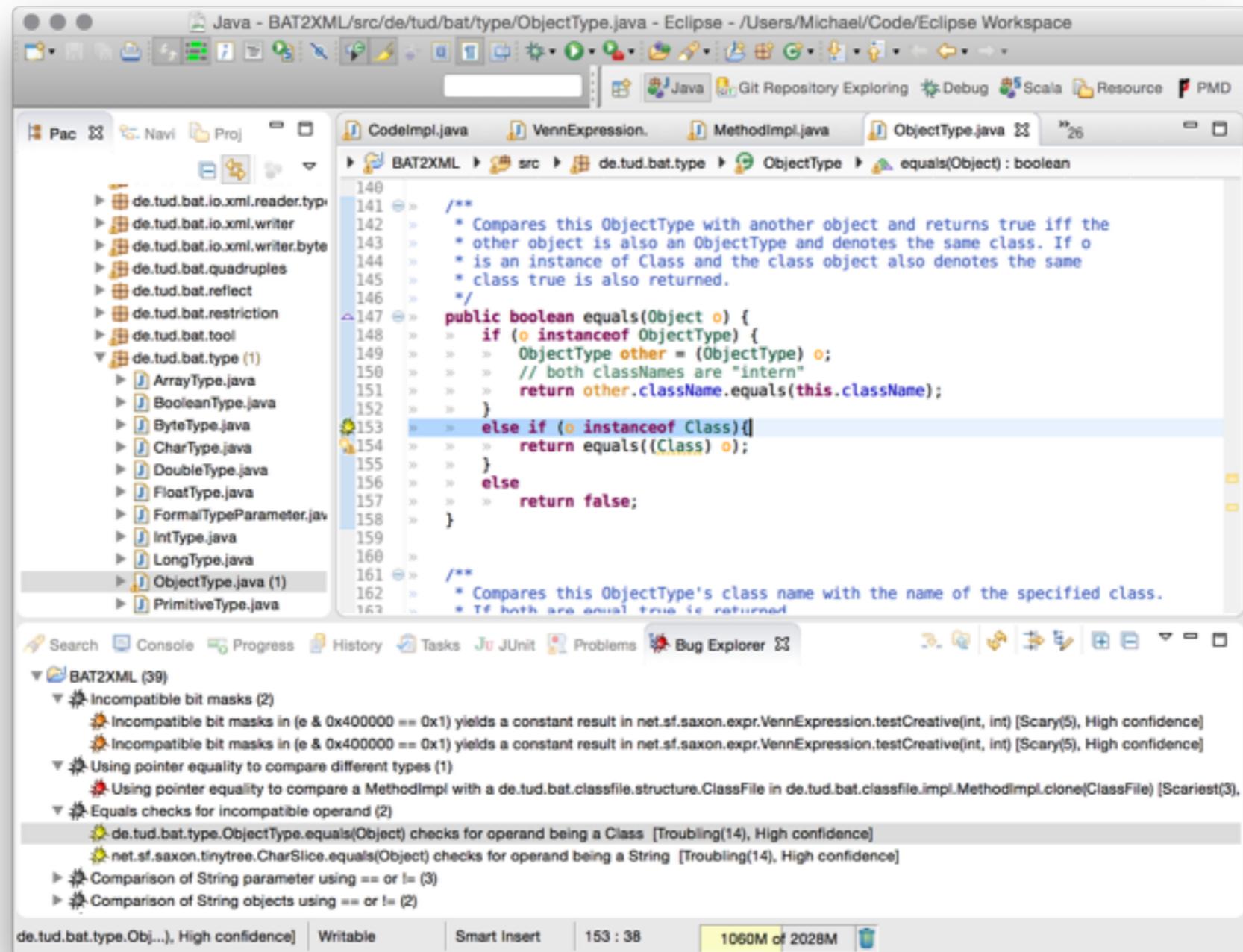
FindBugs Swag

Development

- [Open bugs](#)
- [Reporting bugs](#)

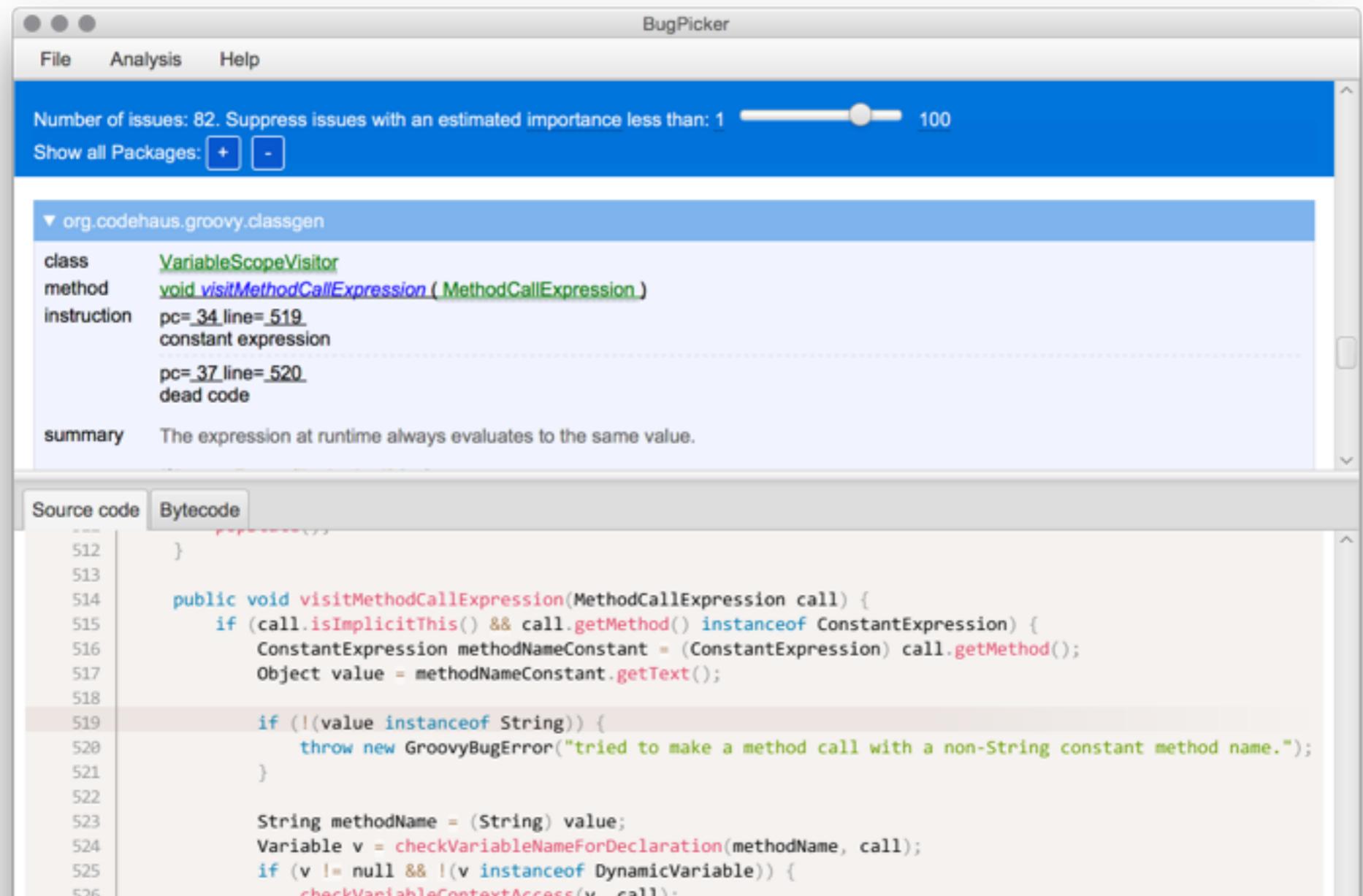
Findbugs

Searches for Bug Patterns in Java Bytecode.
Very good tool integration.



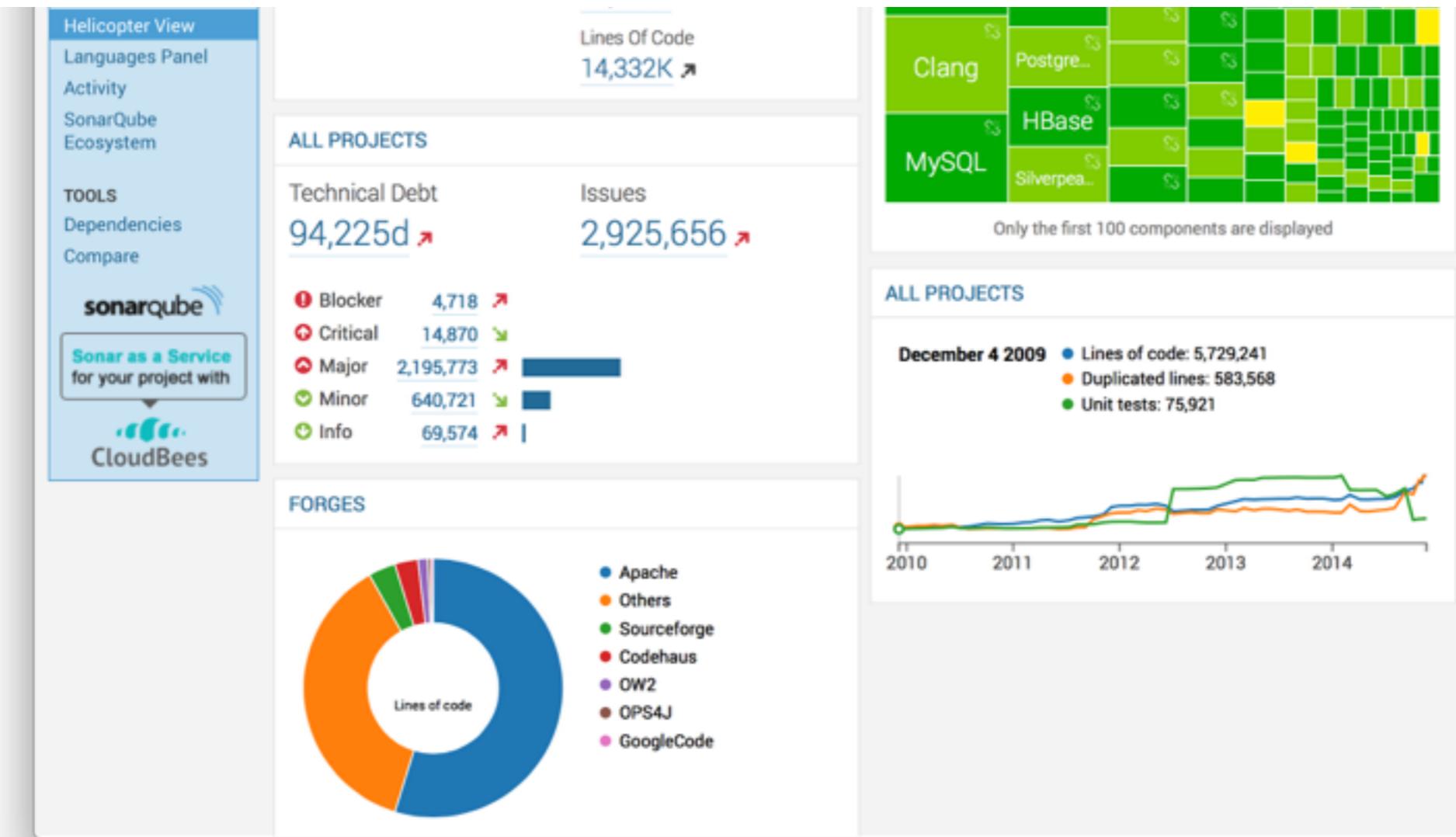
Findbugs

Searches for Bug Patterns in Java Bytecode.
Very good tool integration.



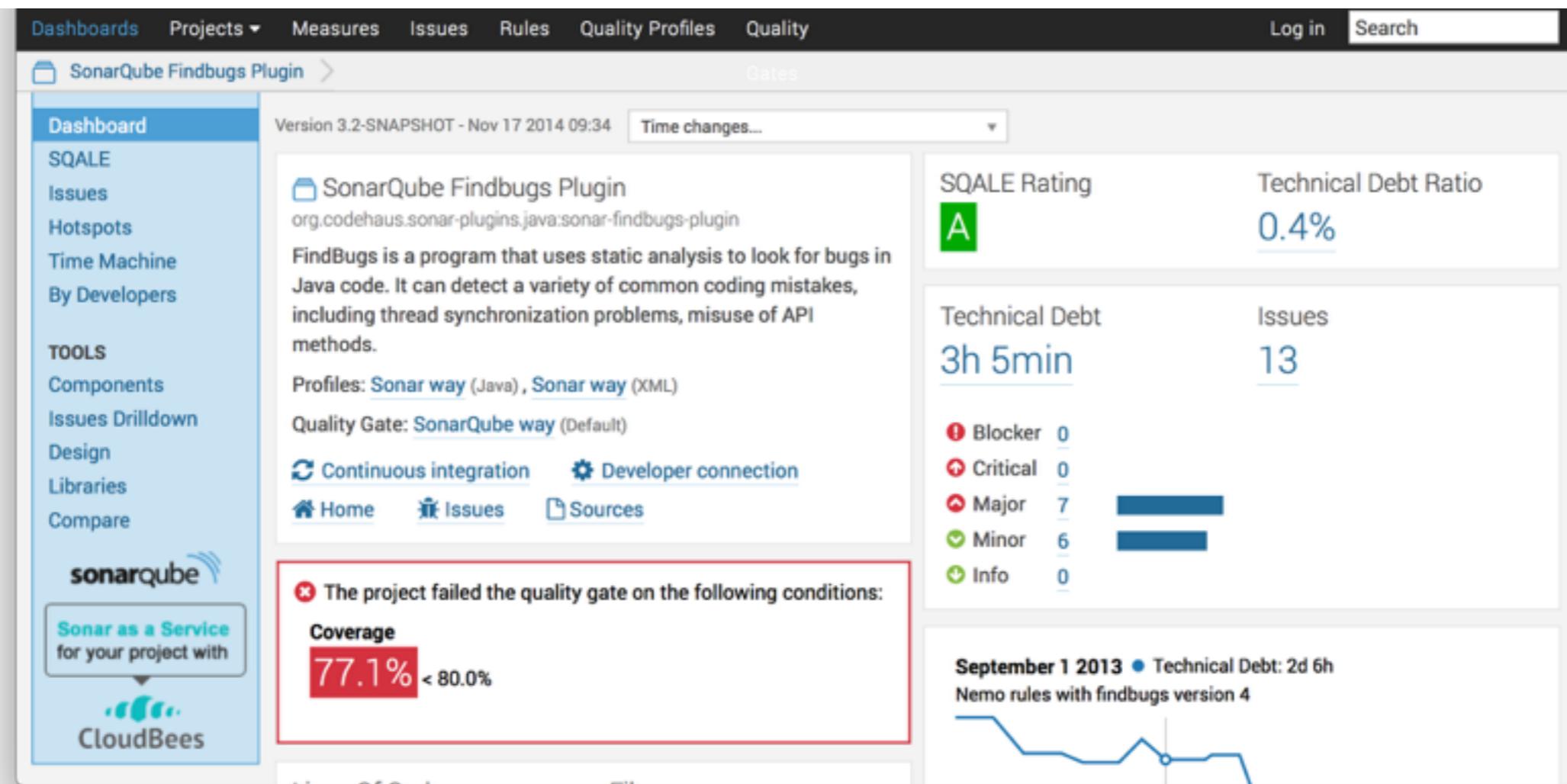
BugPicker

Advanced Dataflow Dependent Bugs



SonarQube

SonarQube is an open platform to manage code quality.



SonarQube

Verschiedene Visualisierungen (Metriken), um die Qualität des Codes sicherzustellen

SonarQube™ technology is powered by SonarSource SA
Version 4.5.1 - [LGPL v3](#) - [Community](#) - [Documentation](#) - [Get Support](#) - [Plugins](#) - [Web Service API](#)

The screenshot shows the SonarQube web interface. On the left, there's a sidebar with links like Dashboard, SQALE, Issues, Hotspots, Time Machine, By Developers, TOOLS, Components, and Issues Drilldown. A banner for 'sonarqube' and 'CloudBees' is also present. The main area has sections for 'Severity' (Blocker: 4,718, Critical: 14,870, Major: 2,195,773, Minor: 640,721, Info: 69,574) and 'Rule' (with several items listed). Below these are lists for 'JDK 7', 'Jetty :: Project', 'Apache CXF', 'Jetspeed-2 Enterprise Portal', 'Apache PhotArk', 'Hudson', and 'testing-main-module'. At the bottom, there's footer text about the technology being powered by SonarSource SA and links to the LGPL v3, Community, Documentation, Get Support, Plugins, and Web Service API.

SonarQube

Integration von Tools zum Finden von “Issues”
(PMD, CheckStyle, FindBugs)

Static Analysis Tools

A FEW (WELL KNOWN)

- FindBugs
Lightweigt static analyses on top of Java Bytecode.
- PMD
Lightweight static analyses on top of the AST using Java Visitors or XPath based rules.
- CheckStyle
Lightweight static analyses on top of the AST using Java Visitors.
- CheckerFramework
Static analyses using pluggable types.
- ConQAT
Code Clone Detection.
- Julia
Static anaylsis of Java and Android code.

Static Analysis Tools

A FEW (WELL KNOWN)

- JDepend
Structural analysis on top of Java Bytecode.
- DependencyFinder
Structural analysis on top of Java Bytecode.
- Stan4J
Structural analysis on top of Java Bytecode.
- Sonargraph (SonarJ)
Analyzes the structure of applications.

Static Analysis Tools

A FEW (WELL KNOWN)

- ESC/Java2
Formal verification using JML Annotations.
- Key
Formal verification.
- ...



FEATURES PRICING DOCS ABOUT US BLOG LOGIN ▾

Continuous Delivery, Containerized

Streamline your development and reduce your test lab footprint



github



bitbucket

Signup for free, no credit card required

Shippable

“Continuous Delivery, Containerized

Streamline your development and reduce your test lab footprint”

Shippable Configuration (Example)

Works for Projects on GitHub or BitBucket

```
reset_minion: true
language: scala
scala:
  - 2.11.8
jdk:
  - oraclejdk8
notifications:
  email:
    - eichberg@informatik.tu-darmstadt.de
```

delors / OPAL



An open, extensible library for the static analysis of Java bytecode.

[Build History](#)[Pull Requests](#)[Notifications](#)

Build History

Page: 1 [▶](#)

Build Group	Status	Triggered	Duration	Changeset	Branch	Committer	Actions
+ 118	success	Today at 12:50 PM	12 minutes	7622f5e	master	Michael Eichberg	
+ 117	success	Today at 10:23 AM	8 minutes	5c48f82	master	Michael Eichberg	
+ 116	success	Yesterday at 3:21 PM	7 minutes	0e8616f	master	Michael Eichberg	
+ 115	success	Yesterday at 2:33 PM	6 minutes	15230dd	master	Michael Eichberg	

Badge

[build shippable](#)

Queued/Running

No Queued/Running Builds

Permissions

1



The Build History

Shippable

118.1

Project	delors/opal
Started at	3 hours ago
Duration	12 minutes
Allow Failure	false

Branch	master
Commit SHA	7622f5e
Matrix Values	runtime=2.11.2 jdk=oraclejdk8

Image	shippable/minv2
Committer	delors
Pull Request	false
Commit Message	the bugpicker now shows all lines associated with an issue report Signed-off-by: Michael Eichberg <mail@michael-eichberg.de>

[Console](#) [Tests](#) [Coverage](#) [Script](#)1759
Passing4
Failures0
Errors34
Skippedclass org.scalatest.exceptions.TestFailedException: expected: MetalInformationUpdate; actual:
NoUpdate

Build Report

Shippable

The screenshot shows the Travis CI interface for the repository `angular/angular.js`. The repository name is at the top left, followed by a search bar and navigation links for Home, Blog, Status, Help, and Sign in with GitHub. Below the search bar is a sidebar titled "Recent" containing a list of other repositories with their names, commit counts, and build times.

The main content area displays the repository `angular/angular.js` with the description "HTML enhanced for web apps". It shows a build for the `master` branch that has just started, indicated by a yellow button labeled "#15292 started" and the text "running for 8 sec". The commit message is: "fix(\$filter): add int support for negated strict comparison". The commit was authored by Adi Chikara and committed by the same user. The commit hash is `650ddda`.

Below the commit details is a "Build Matrix" table:

Job	Duration	Finished	Node.js	ENV	OS
15292.1	8 sec	-	0.10	JOB=unit	linux
15292.2	8 sec	-	0.10	JOB=e2e TEST_TARGET=jqlite	linux
15292.3	-	-	0.10	JOB=e2e TEST_TARGET=jquery	linux

Travis CI

A hosted continuous integration service for open source and private projects.